

EXHIBIT 9

DOCUMENT SOUGHT TO BE SEALED

UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

_____)	
ORACLE, INC.)	
)	
Plaintiff,)	
)	
v.)	Case No. CV 10-03561 WHA
)	
GOOGLE INC.,)	
)	
Defendant.)	
_____)	

HIGHLY CONFIDENTIAL

EXPERT REPORT OF DR. GREGORY K. LEONARD

CORRECTED (MARCH 10, 2016)

65. In a second step, the Android profit figure of \$1.9 billion is itself apportioned to all of the various functionalities of Android and Android handsets, of which the allegedly infringing material constitutes only a small part, as described below.

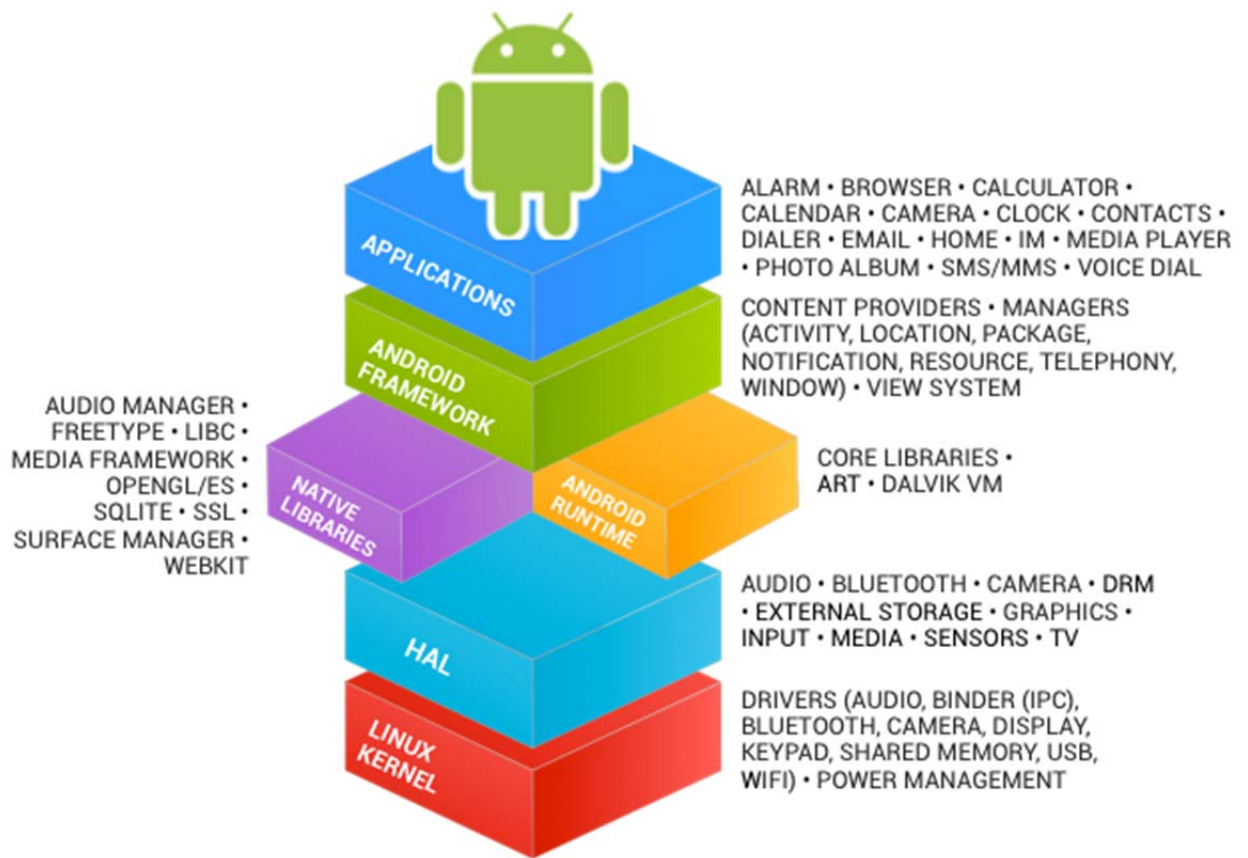
C. Reasons for the Success of Android Other Than the Alleged Infringement

1. Google's Efforts

66. Google has been the main driver behind the initial launch and continued development of Android. Android launched as version 1.0 and is currently on version 6.0.⁹⁰

67. As pictured below, Android consists of a Linux kernel, a hardware abstraction layer, native libraries, Android runtime, the Android framework, and a set of built-in applications.

⁹⁰ For my understanding of the technical aspects of Android described in this section, I rely on the expert reports and testimony of Google's technical experts in this phase and in the previous trial. I understand that the code related to the 37 API packages at issue underwent little change after Android's launch. Kemerer 1/8/2016 Report ¶ 110. Therefore, all or nearly all of the value Google added to Android in versions subsequent to 1.0 involved functionality other than the 37 API packages.



Source: <https://source.android.com/source/index.html>

a. Linux Kernel

68. The Linux kernel is the portion of the Android operating system that interacts with and runs the hardware—the physical phone or tablet used by the consumer, including the camera, speakers, and display screen. Linux is available on an open source basis. However, “Android uses a version of the Linux kernel with a few special additions such as wake locks (a memory management system that is more aggressive in preserving memory), the Binder IPC driver, and other features important for a mobile embedded platform.”⁹¹ In other words,

⁹¹ “Android Interfaces and Architecture,” Android, <https://source.android.com/devices/index.html>.

Google has made contributions to the open-source Linux kernel that provide additional, customized functionality for Android.⁹²

b. Hardware Abstraction Layer

69. The hardware abstraction layer (HAL) provides an interface between the software APIs and the hardware. Specifically, it “defines a standard interface for hardware vendors to implement and allows Android to be agnostic about lower-level driver implementations. The HAL allows you to implement functionality without affecting or modifying the higher level system.”⁹³

c. Native Libraries, Core Libraries, and the Android Framework

70. The native libraries, core libraries, and Android Framework provide APIs that can be accessed by applications that run on Android.⁹⁴ The native libraries are written in the C++ programming language, not in Java. The core libraries are “Java” based, although some of the “Java” APIs are just “wrappers” around a C++ API. In other words, when one of these APIs is called, the computational work is done by C++, not Java. The Android Framework provides an easy-to-work-with environment for developers.⁹⁵

71. The 37 API packages at issue in this case are part of the core libraries. However, Google provided over 131 other API packages to the Android native libraries, core libraries, and Android Framework. Google has continued to innovate, continually adding large numbers of

⁹² “Android A to Z: What is a Kernel? Android Central, January 23, 2012, <http://www.androidcentral.com/android-z-what-kernel>.

⁹³ “Android Interfaces and Architecture,” Android, <https://source.android.com/devices/index.html>.

⁹⁴ Trial Tr. 2161:10-2171:22 (Astrachan).

⁹⁵ Interview of Tim Bray; “NARRATIVE: Mobile + Android, last updated: 03/18/08,” GOOGLE-23-00000001-027 at 003. Also see “Android 101: An Introduction to Android and Android Partnerships, Last Updated: December 2008,” GOOGLE-00298438-484 at 466.

features to Android and corresponding APIs. For example, with Lollipop, Google added approximately 3,000 new APIs.⁹⁶

d. Android Run Time

72. Applications are computer programs that are compiled and then run on an Android device. Initially, Google designed Android so that each application written in the Java programming language runs in what was called the “Dalvik” virtual machine. I understand that the Dalvik virtual machine is not subject to any of the copyrights asserted in this case. Rather, aspects of Dalvik were accused by Oracle in the first trial of this case of infringing certain Oracle patents, but Google prevailed as to those issues, obtaining a verdict of no infringement from the jury. Dalvik was designed to be efficient and compact,⁹⁷ which can be important given the mobile device context.

73. Google subsequently developed “Android Run Time” (ART) to replace Dalvik. Under ART, applications are compiled to machine code upon installation. This provides benefits, including improved performance of the application.⁹⁸

e. Applications Layer

74. Android provides a large number of functionalities through built-in applications. These include the software that runs a web browser, email, calendar, phone, text messaging, camera, calculator, and flashlight, among others. These built-in applications provide users with a substantial amount of functionality even in the absence of any third party apps.

⁹⁶ Lockheimer 12/8/2015 Dep., p. 271:5-15.

⁹⁷ Ghuloum 12/9/2015 Dep., pp. 155:2-156:24.

⁹⁸ “ART and Dalvik,” Android, <https://source.android.com/devices/tech/dalvik/>.

trouble adapting to the mobile environment, again imposing a negative externality on Android.¹⁶⁹ In general, young programmers creating the most interesting applications do not like the Java programming language because it is considered obsolete.¹⁷⁰

112. The disadvantages to Java have contributed to the fact that many applications available on both iOS and Android were launched first on iOS. According to a 2010 Nielsen study, Facebook, Pandora, and the Weather Channel are among the most popular apps (as measured by number of recent downloads) on both Android and iPhone.¹⁷¹ Each of these apps was launched on iPhone before it was launched on Android. Facebook was launched on iOS in July 2008, as compared to July 2009 on Android; Pandora was launched on iOS in July 2008, as compared to September 2009 on Android; Weather Channel was launched on iOS and Android simultaneously.¹⁷² Similarly, among the list of “Top 30 Android Apps of All Time” as determined

¹⁶⁹ Interview of Tim Bray.

¹⁷⁰ Interview of Tim Bray.

¹⁷¹ “The State of Mobile Apps,” Nielsen, June 1, 2010, <http://www.nielsen.com/us/en/insights/news/2010/the-state-of-mobile-apps.html>.

¹⁷² “Facebook for iPhone Application Launches,” Social Times, July 10, 2008, <http://www.adweek.com/socialtimes/facebook-for-iphone-application-launches/212245>; “The Weather Channel Tops 10 Million iPhone App Downloads,” The Weather Company, April 28, 2010, <http://www.theweathercompany.com/newsroom/2014/08/19/weather-channel-tops-10-million-iphone-app-downloads>; “The Weather Channel App for Android Launches Major Updates,” Lost Remote, February 28, 2013, <http://www.adweek.com/lostremote/the-weather-channel-app-for-android-launches-major-updates/37856>; “Happy Birthday to the Pandora App,” Pandora Blog, July 10, 2013, <http://blog.pandora.com/2013/07/10/happy-birthday-to-the-pandora-app/>; “Pandora for Android,” Pandora Blog, September 9, 2009, http://blog.pandora.com/2009/09/09/pandora_for_and/.

by a 2010 TechCrunch article, 15 are non-Google apps that are also on iPhone.¹⁷³ Eleven of the 15 apps were launched on iPhone before they were launched on Android.¹⁷⁴

113. The fact that developers tended to write their apps for iOS before Android demonstrates two points. First, these developers were not deterred in developing their apps for iPhone despite having to write the apps in Objective C, which was historically a little-used programming language and remains relatively unpopular today despite Apple's adoption of it

¹⁷³ The other 15 apps are Google-developed apps, utilities that are specific to the Android OS (such as an anti-virus app), or games exclusive to Android. "Top 30 Android Apps of All Time," TechCrunch, October 30, 2010, <http://techcrunch.com/2010/10/30/top-30-android-apps/>.

¹⁷⁴ "Kindle for Mac Now Finally Available," Engadget, March 18, 2010, <http://www.engadget.com/2010/03/18/kindle-for-mac-now-finally-available/>; "Amazon Releases Kindle App for Android Phones," New York Times, June 28, 2010, <http://bits.blogs.nytimes.com/2010/06/28/amazon-kindle-app-now-available-for-android/?pagemode=print>; "Evernote News," Evernote, July 10, 2008, <https://evernote.com/corp/news/pr/2008-07-10.php>; "Evernote for Android: It's Here!" Evernote, December 16, 2009, <https://blog.evernote.com/blog/2009/12/16/evernote-for-android-its-here/>; "Facebook for iPhone Application Launches," Social Times, July 10, 2008, <http://www.adweek.com/socialtimes/facebook-for-iphone-application-launches/212245>; "Facebook Launches Official Google Android Application," Mashable, September 8, 2009, <http://mashable.com/2009/09/08/facebook-android/#MUlofPT.rsqz>; "Becomes No. 1 iTunes Finance Application Within 24 Hours," Mint, December 22, 2008, <https://www.mint.com/press/mint-introduces-free-iphone-application>; "Mint Comes to Android," Readwrite, May 3, 2010, http://readwrite.com/2010/05/03/mint_comes_to_android; "Facebook for iPhone Application Launches," Social Times, July 10, 2008, <http://www.adweek.com/socialtimes/facebook-for-iphone-application-launches/212245>; "Sing for Search Results with iPhone App," CNET, July 11, 2008, <http://www.cnet.com/news/sing-for-search-results-with-iphone-app/>; "Dropbox for iPhones Is Out (and Awesome)," Mashable, September 29, 2009, <http://mashable.com/2009/09/29/dropbox-iphone/#to8U3hEnrZqW>; "The Dropbox Android App," Dropbox Blog, March 22, 2010, <https://blogs.dropbox.com/dropbox/2010/03/the-dropbox-android-app-2/>; "First Look: LogMeIn Ignition," Engadget, January 19, 2009, <http://www.engadget.com/2009/01/19/first-look-logmein-ignition/>; "LogMeIn Ignition Launches Today, \$29.99 in the Android Market," Phandroid, July 14, 2010, <http://phandroid.com/2010/07/14/logmein-ignition-launches-today-29-99-in-the-android-market/>; "Fruit Ninja," Gamespot, April 20, 2010, <http://www.gamespot.com/fruit-ninja/>; "Fruit Ninja now available on Android!," Half Brick, September 17, 2010, <http://halfbrick.com/fruit-ninja/fruit-ninja-on-android/>; "HomeRun Battle 3D," IGN, June 17, 2009, <http://www.ign.com/games/home-run-battle-3d/iphone-14356201>; "Homerun Battle 3D goes Goid, er... Droid," TouchMyApps, January 7, 2010, <http://www.touchmyapps.com/2010/01/07/homerun-battle-3d-goes-goid-er-droid/>; "The Best iPhone Apps of 2009," Tech Crunch, December 27, 2009, <http://techcrunch.com/2009/12/27/best-iphone-apps-2009-appvee/>; "Zenonia and other Top Android Games of the Week," App Olicious, August 16, 2010, <http://www.appolicious.com/articles/2794-zenonia-and-other-top-android-games-of-the-week>; "Rovio's 'Angry Birds 2' to launch on Apple's iOS July 30," Apple Insider, July 16, 2015, <http://appleinsider.com/articles/15/07/16/rovios-angry-birds-2-to-launch-on-apples-ios-july-30>; "Angry Birds" Now Available on Android for Free," Mashable, October 15, 2010, <http://mashable.com/2010/10/15/angry-birds-android-2/#EmCWRAtkyqqF>.

for use with iOS devices. In fact, some developers prefer Objective C to Java given its greater efficiency and more up-to-date libraries.¹⁷⁵ Second, each company delayed developing its app for Android despite being able to write the app in Java, most likely to wait for Android to demonstrate sufficient user growth to show that Android offered a market opportunity worth exploiting by developers (a topic discussed in greater detail below). This demonstrates that a platform's revenue potential is a much more important driver of developer choice to develop for the platform than is its applications programming language.¹⁷⁶

114. If Oracle is correct that Google's use of Java as an applications programming language in Android made app development easier due to developer familiarity with Java, we should observe that, of the developers that dual-homed (i.e., offer their app on both iPhone and Android), those that developed their Android apps in Java exhibited a shorter lag between the iPhone and Android launches of their apps than those that developed their Android apps in C++ (using the NDK). This hypothesis can be tested by running a regression of a developer's iPhone/Android launch lag on an indicator variable for whether the Android app was written in Java. Effects of relative installed base can be controlled for by also including indicator variable for time of the app's launch as explanatory variables.

115. From App Annie data, I identified apps appearing on Top 100 monthly download lists during the 2012-2013 period that multi-homed on both the Android and iPhone platforms.

¹⁷⁵ "Reservoir Devs: Why Apps Really Hit iOS Before Android," December 17, 2015, <http://thenextweb.com/dd/2015/12/17/reservoir-devs-why-apps-really-hit-ios-before-android/#gref>. This article summarized responses of developers who were asked about their platform choices. One developer stated, "There are so many incredible third-party libraries made with Swift and Objective-C now that in my mind developing for Android is like being in a fight with one hand tied behind your back."

¹⁷⁶ Ghuloum 12/9/2015 Dep., pp. 148:12-149:3.

Python, and C/C++.¹⁷⁸ Even after the iPhone's success was established and a large number of applications had been developed for the iPhone platform, Objective C was still ranked well below other "top tier" languages.¹⁷⁹ Yet, despite the lower levels of familiarity with Objective C, there has been an explosion in applications developed for the iPhone.¹⁸⁰ This shows that developers are willing and able to respond to market opportunities by adapting and learning to program in previously unfamiliar languages.

118. Even after Android surpassed iPhone in terms of shipments and user base, it was still accepted as common wisdom in the industry that "developers should write for iOS (Apple) first and then Android (Google) second."¹⁸¹ This is not consistent with Android having a substantial advantage over iOS due to using Java as an applications programming language.

119. In 2014, Apple introduced an entirely new programming language called Swift as an applications programming language for iOS devices, with the ultimate intention of replacing Objective C in that role. Despite being entirely new, Swift has quickly gained acceptance and iOS developers have switched over from Objective C in great numbers, because of its adoption by Apple, an established and successful smartphone-platform owner, and the size and relative certainty of the market opportunity offered to developers willing and able to learn and program

¹⁷⁸ According to the TIOBE index cited by Oracle's experts, Objective C was ranked low in popularity. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. This is still the case. A 2015 survey of developers (not restricted to mobile app developers) found that only 36% of respondents were moderately or highly conversant with Objective C as compared to 68% for Java. Developer Insights Report, IDC, August 2015, p. 26. This study notes that Java number has benefited from Android, so the two numbers would have been closer at the time of Android's launch. Id.

¹⁷⁹ Id.

¹⁸⁰ "Mobile Developer Economics 2010 and Beyond," VisionMobile, July 2010, p. 10.

¹⁸¹ T. Bresnahan, J. Davis, and P.L. Yin, "Economic Value Creation in Mobile Applications," in *The Changing Frontier: Rethinking Science and Innovation Policy* (A. Jaffe and B. Jones, eds., 2015) (hereafter "Bresnahan, et al. (2015)"), at location 5838 (Kindle version).

for both the Android and iOS operating systems. The results of the survey are consistent with and confirm the conclusions of the above discussion, including:²⁴¹

- The size of a mobile platform's user base and the expected profitability of developing for that user base are the most important considerations in developers' decisions to develop for a platform, while familiarity with the programming language is a minor consideration at best for only a small number of developers (and even then, a far less important consideration compared to the others).
- There is no reason to believe that developers would decline the opportunity to develop for a new mobile platform that showed market potential (for example, through a growing base of users) because of the programming language used by that platform.
- Many Android developers also develop for iOS, which suggests that many developers know and work using different programming language and different applications programming environments, demonstrating the flexibility to respond to market conditions and opportunities.
- Developers are confident that they could learn a new programming language, with many in fact first learning Objective C when they started developing for iOS and then learning or planning to learn Swift, the new iOS programming language recently introduced by Apple.

d. Growth in Android Apps Followed the Growth in the Number of Android Users, Not Vice-Versa

163. In 2009 and 2010, Android app development lagged behind iPhone app development, despite the fact that the Android user base was growing.²⁴² Developers were hesitant to initiate Android app development before there was a return that could justify the cost, and this break-even return depended in part on the existence of a sufficiently large user

²⁴¹ See generally Simonson 2/8/2016 Report.

²⁴² See, e.g., "App Makers Take Interest in Android," New York Times, October 25, 2010, http://www.nytimes.com/2010/10/25/technology/25android.html?_r=0 ("Even as Android sales surge – Google says it is now activating around 200,000 phones per day – the market for Android apps still seems anemic compared with that for Apple and its thriving App Store"); "The Rise of Android, A Deep Dive Analysis," Caris & Company, September 13, 2010.

base.²⁴³ As a recent economic study concludes, “the rapid emergence of many demanders, together with the very low barriers to entry created by the platform providers, has led to a rapid and very substantial expansion in the number of overall apps.”²⁴⁴ Google’s Mr. Rutledge testified that in the 2008 to 2009 time frame, Google “had quite a hard time getting any interest of any application developer to go for Android...Primarily [because of] the size of the device ecosystem. We had no market share.”²⁴⁵

164. Android was initially launched in October 2008 on a single HTC handset and at a single carrier in the U.S., T-Mobile, which at the time was only the fourth largest U.S. carrier with a 12% share of the mobile phone market.²⁴⁶ Given the limited distribution potential, the Android installed base was still small one year following the launch of the first Android handset.²⁴⁷ Also contributing to the low level of installed base were limitations of the HTC handset and Android 1.0 itself.²⁴⁸

²⁴³ Meier 12/11/2015 Dep., pp. 43:10-46:24. Mr. Meier noted that large companies, in particular, could afford to wait until Android had built a user base before investing in an Android app (pp. 52:19-53:24). Yet, such companies are those that provide some of the most popular apps as discussed elsewhere in this report.

²⁴⁴ Bresnahan, et al. (2015), at location 5883 (Kindle version).

²⁴⁵ Rutledge 12/9/2015 Dep., pp. 77:17-23. See also Rutledge Dep. Ex. 5057 (“Developers need to see volume shipments before they write apps.”); Rutledge 12/9/2015 Dep., p. 109:03-110:10; Meier 12/11/2015 Dep., pp. 40:8-41:21; “Additional Item: If Android Does Succeed It Will Be By Quite a Narrow Margin,” Optical Networks Daily, August 7, 2009.

²⁴⁶ “A Brief History of Android Phones,” CNET, August 2, 2011, <http://www.cnet.com/news/a-brief-history-of-android-phones/>; Fourteenth Annual Report and Analysis of Competitive Market Conditions With Respect to Mobile Wireless, Including Commercial Mobile Services, Federal Communications Commission, May 20, 2010, p. 31.

²⁴⁷ Exhibit 2h.

²⁴⁸ Regarding limitations of the handset, see “T-Mobile G1 review,” CNET, October 15, 2008, <http://www.cnet.com/products/t-mobile-g1/> (“for now, the [HTC handset] is best suited for early adopters and gadget hounds, rather than consumers and business users”). When Android 1.5 was released, it was viewed as a major improvement over 1.0, demonstrating the limitations of 1.0. See “Top 10 Features You’ll Love about Android 1.5,” <http://www.geek.com/android/top-10-features-youll-love-about-android-15-768061/> (1.5 “significantly enhance[s] the [user] experience, addressing most glaring omissions from the 1.0 release”);

165. The limited initial availability of Android devices, and the resulting slow initial growth of the Android user base, also are inconsistent with the suggestion by Oracle's experts Dr. Kemerer and Mr. Malackowski that Google faced a limited "window" of opportunity to launch Android, and that if Google missed that window, Android would have failed. Oracle's experts use this suggestion to argue, in essence, that had Google not used the allegedly infringing works in this case, Android would not exist today, or in any event that Google would have earned substantially less in profit from Android than it has. However, I note that none of Oracle's experts provide any economic analysis establishing that such a "window" existed, much less showing when it began, when it ended, and its significance. Oracle's experts rely on contemporaneous Google documents discussing Google's perceptions of the market opportunity, but do not appear to have done any analysis to verify whether and to what extent Google's perception was accurate or that the window would have soon closed. In fact, to the extent there was any such window, it was reasonably wide in terms of time, as demonstrated by the fact that, even a year after the Android launch, the Android user base was still quite small, owing in part to the limited number of Android devices in the market with smaller carriers. Yet, the Android user base grew substantially in subsequent years, particularly after the introduction of additional, and higher-end, Android devices with the larger carriers

"Google Phone Update: Android 1.5 'Cupcake' Reviewed," http://www.informationweek.com/mobile/mobile-devices/google-phone-update-android-15-cupcake-reviewed/d/d-id/1080104?page_number=1 ("In sum, the 1.5 update to Android makes the platform much stronger. It adds a host of missing features, makes improvements to the overall usability of the operating system, and makes the platform feel more complete"); "Google Android 1.5," <http://www.pcmag.com/article2/0,2817,2349058,00.asp> ("The latest version of the OS, Android 1.5, features a number of major enhancements").

beginning in late 2009, two years after the introduction of the Android platform. Thus, the “window” had not closed within at least two years after Android’s actual introduction.

166. Just prior to Android’s launch, Google removed some apps from the Android app store, which reduced the number of apps in the store to 13.²⁴⁹ The number of new apps added to the store per month remained at low levels during the first year after the Android launch.²⁵⁰

167. In October 2009, the distribution potential for Android increased considerably when Samsung and Motorola introduced their first Android handsets in the U.S. and these handsets were available with the carriers Sprint and Verizon, which were the third and first largest carriers in the U.S.²⁵¹ Moreover, these handsets were well-received and the Motorola Droid had the newest version of Android, Android 2.0 (the Samsung handset had Android 1.5, which was still a major improvement to Android 1.0).²⁵² The Android installed base picked up after this point in time, and this in turn was followed by a jump in the number of new apps per month.²⁵³

²⁴⁹ “Google Removes Applications Just Before Launch,” Android Community, <http://androidcommunity.com/google-removes-applications-just-before-launch-20081020/>. Shortly after the Android launch, reports suggest that there were about 62 apps in the Android app store. “Top Ten Android Launch Apps,” Tech Crunch, <http://techcrunch.com/2008/10/22/top-ten-android-launch-apps/>.

²⁵⁰ Exhibit 2g.

²⁵¹ “A Brief History of Android Phones,” CNET, August 2, 2011; Sixteenth Annual Report and Analysis of Competitive Market Conditions With Respect to Mobile Wireless, Including Commercial Mobile Services, Federal Communications Commission, March 21, 2013, p. 54.

²⁵² “Motorola Droid (Verizon Wireless) review,” CNET, October 28, 2009, <http://www.cnet.com/products/motorola-droid-verizon-wireless/>; “First Google Android 2.0 Phone Arrives,” CNET, October 30, 2009, <http://www.cnet.com/news/first-google-android-2-0-phone-arrives/>; “Top 10 Features You’ll Love about Android 1.5,” Geek, May 8, 2009, <http://www.geek.com/android/top-10-features-youll-love-about-android-15-768061/>; “First Look – Samsung I7500 Preview,” GSMArena, May 21, 2009, http://www.gsmarena.com/samsung_i7500-review-351.php; “Google Phone Update: Android 1.5 ‘Cupcake’ Reviewed”, InformationWeek, June 2, 2009, http://www.informationweek.com/mobile/mobile-devices/google-phone-update-android-15-cupcake-reviewed/d/d-id/1080104?page_number=1.

²⁵³ Exhibits 2g, 2h.

168. This is an example of how, as discussed in greater detail below, the growth in Android users was driven to a large degree by improvements in the price-adjusted quality of the devices running Android and additional functionalities provided by Android itself rather than the available apps. These hardware and OS improvements reflect the contributions of the OEMs and Google as well as the open source strategy of Google that resulted in a low price for the OS and thus encouraged OEMs to adopt Android.

e. Economics Literature on Platform Choice by App Developers and Smartphone Choice by Users

169. Several recent studies have examined platform choice by mobile app developers. A 2015 study by T. Bresnahan, et al. examines issues related to platform choices of app developers. Their empirical results are “consistent with the theory that the most important determinant of developer behavior is the installed base.”²⁵⁴ They find that many developers write apps for both Android and iOS, which is consistent with the conclusion that “the technical entry barriers to either platform are low for developers.”²⁵⁵ The technical entry barriers would include the software development and thus, in principle, could be affected by the choice of programming language. However, the fact that the technical entry barriers are low suggests that any such effect is minimal. The study itemizes a number of “asymmetries” between the Android and iOS platforms, one of which is that Java is “popular with developers”; however, the authors focus on Android’s openness as the primary important advantage over iOS.²⁵⁶ Finally,

²⁵⁴ Bresnahan, et al. (2015), at location 6527 (Kindle version).

²⁵⁵ Id. at location 6527 (Kindle version). The authors conclude that the marketing costs, however, are an important entry to barrier for apps. For example, getting on top app lists is both important for generating demand and costly. Such marketing costs are not related to the specific applications programming language.

²⁵⁶ Id. at locations 6527, 6557 (Kindle version).

assuming a benefits to salary percentage of 31.7%, the incremental cost would be \$84,722. The total cost savings, and thus unjust enrichment (assuming infringement), that Google received from using the allegedly infringing material is approximately \$85,000.²⁷²

b. Avoidance of Costs Associated With Developer Training

179. Another measure of the contribution of the alleged infringement is the cost that Google would otherwise have had to incur to train third party app developers who were unfamiliar with an alternative programming language, such as C or C++. Note that no developer training would be needed if Google had implemented the OpenJDK class libraries. Thus, developer training cost-savings is an alternative measure of the unjust enrichment (assuming infringement).

180. Based on the App Annie data, I have determined that during the period 2012-2015, there were 1,889 developers that had an Android app in the daily top 100 download list for at least one day.²⁷³ Given the extreme skew in the demand for apps (i.e., a few apps are highly demanded and most are marginal; 200 apps account for the large majority of usage, etc.), this list captures the developers of important Android apps. Some of these developers used C/C++ to develop their apps and thus would not need to be trained in C/C++. I have used a list of the Java/NDK status of apps provided by Google to eliminate developers that knew C/C++ already (this is a conservative approach given that this Java/NDK status list did not include all apps; if an app is not on the list, I assume it is Java). I also eliminated developers that

²⁷² Exhibit 3a.

²⁷³ The market intelligence data from App Annie provides daily ranking, number of downloads, and developer information for the top 100 free and paid apps in iOS and Google Play between January 2012 and December 2015. The data also identifies the different versions of an app across and within platforms. "App Annie Intelligence Product Suite Overview," App Annie.

multi-homed on iOS and thus had demonstrated an ability to develop in multiple languages. After those two sets of developers are eliminated, 986 developers are left. Some of these developers may well have already known C/C++ or been willing to learn another language without being paid by Google. However, to derive a conservative estimate of the savings in training costs, I assume all of these developers would need to be trained. I double this number to account for the entire 2008-2015 time period. Finally, assuming 1.6 programmers per developer on average²⁷⁴, there would be a maximum of 3,155 programmers that would need to be trained in C/C++. There are many C/C++ courses available for free; as discussed above, paid courses cost approximately \$715.²⁷⁵

181. Thus, a conservative measure of the cost-savings for Google from having to avoid training developers in another programming language such as C/C++ is \$2,256,000.²⁷⁶

c. Avoidance of Costs Associated With Paying Third Party Developers to Develop Android Apps

182. Another measure of the contribution of the alleged infringement is the cost that Google avoided having to incur to promote app development by third party developers. Note that paying for app development would not be needed if either Google had implemented the OpenJDK class libraries or Google had provided training to developers unfamiliar with C/C++.

²⁷⁴ There were a total of 73 developers working on the 46 applications made public in the list of Top 50 Android Applications for Google's Developer Challenge in 2008. This gives an average of 1.6 developers per app. See "Google Reveal The Top 50 Android Applications (46 Public)," Chris Moor at Talk Android, <http://www.talkandroid.com/92-developer-challenge-top-50-android-application/>; "Splash Play," AndroidTapp, <http://www.androidtapp.com/splashplay/>.

²⁷⁵ Exhibit 3c.

²⁷⁶ See Exhibit 3c.

VIII. Awarding Lost Profits in Addition to Unjust Enrichment Would Double-Count Damages

283. I understand that a plaintiff asserting copyright infringement may seek recovery of both the unjust enrichment to the alleged infringer and any actual damages suffered by the copyright owner, such as lost profits, as long as the latter are not duplicative of the former.

284. Here, the claimed actual damages are Oracle's lost profits arising from claimed lost Java ME licensing revenues. The claimed lost Java ME licensing revenues, in turn, arise from claimed displacement of Java ME by Android. Thus, the claimed lost Java ME profits are the flip side of the Android profits that Oracle claims as unjust enrichment. Accordingly, it would be duplicative to award both the claimed actual damages consisting of Java ME lost profits and the claimed unjust enrichment.



Gregory K. Leonard
March 10, 2016